Claims

What is claimed is:

1. A method of controlling a computer system running one or more application programs and an operating system incorporating a kernel, said method comprising:

   providing a privileged mode for executing routines associated with said kernel, and a non-privileged mode for executing routines associated with an application program;

   locating the kernel within a first region of memory, and the application program in a second region of memory; and

   setting the system to privileged mode in response to accessing code in said first region of memory and to non-privileged mode in response to accessing code in said second region of memory.

2. The method of claim 1, wherein the application program in the second memory region accesses the kernel in the first memory region by making a system call.

3. The method of claim 2, wherein said system call is implemented as a standard function call.

4. The method of claim 3, wherein said setting comprises:

   switching to privileged mode in response to a function call from the second memory region into the first memory region; and

   switching to non-privileged mode in response to a function call from the first memory region into the second memory region.

5. The method of claim 3, wherein said setting further comprises reverting to a previous mode at the conclusion of a function call.

6. The method of claim 3, wherein said setting further comprises comparing a calling address and a called address for the function call against predetermined address limits to determine in which memory region said calling address and called address are located.

7. The method of claim 1, wherein the computer system memory is divided into pages, and the method further comprises associating an indicator with each page representative of whether the page is located in said first region of memory or said second region of memory.

8. The method of claim 7, wherein said indicator is stored in a page translation table.

9. The method of claim 7, wherein said indicator has a first value for the second memory region, and a second or third value for said first memory region, and wherein:

code on a page having said first value executes in non-privileged mode and can be accessed from code on a page having said first, second or third value;

code on a page having said second value executes in privileged mode and can be accessed from code on a page having said first, second or third value; and

code on a page having said third value executes in privileged mode and can be accessed from code on a page having said second or third value.

10. The method of claim 9, wherein a page having said second value is used to reference code routines on one or more pages having said third value.

11. The method of claim 1, wherein said first memory region is divided into first and second sub-regions, and wherein a function call from the second memory region is permitted into only one of said first and second sub-regions.

12. The method of claim 1, wherein said first and second memory regions are determined as part of system initialisation.

13. The method of claim 1, wherein at least a portion of device driver code is located in the second memory region.

14. The method of claim 1, wherein at least a portion of trusted application code is located in the first memory region.

15. A computer system running one or more application programs and an operating system incorporating a kernel and comprising:

means for providing a privileged mode for executing routines associated with said kernel, and a non-privileged mode for executing routines associated with an application program;

means for locating the kernel within a first region of memory, and the application program in a second region of memory; and

means for setting the system to privileged mode in response to accessing code in said first region of memory and to non-privileged mode in response to accessing code in said second region of memory.

16. A computer system comprising:

one or more application programs having routines that execute in a non-privileged mode;

an operating system incorporating a kernel having routines that execute in a privileged mode; and

a memory, wherein the kernel is located within a first region of memory, and the application program is located within a second region of memory; and

wherein the system is responsive to an access to code in said first region of memory to set the system into privileged mode, and is responsive to an access to code in said second region of memory to set the system into non-privileged mode.

17. The system of claim 16, wherein said access to code in said first region of memory and said access to code in said second region of memory each represents a jump of processing location.

18. The system of claim 17, wherein said access to code in said first region is made by said one or more application programs making a system call.

19. The system of claim 16, wherein the system maintains predetermined address limits for said first and second memory regions, and determines an access to said first and second memory regions by comparing an accessed address with said predetermined address limits.

20. The system of claim 16, wherein the computer system memory comprises multiple pages, and the system further comprises a page translation table having an entry for each page, wherein each entry includes an indicator representative of whether the page is located in said first region of memory or in said second region of memory.

21. The system of claim 20, wherein said indicator has a first value for the second memory region, and a second or third value for said first memory region, and wherein:

code on a page having said first value executes in non-privileged mode and can be accessed from code on a page having said first, second or third value;

code on a page having said second value executes in privileged mode and can be accessed from code on a page having said first, second or third value; and

code on a page having said third value executes in privileged mode and can be accessed from code on a page having said second or third value.

22. The system of claim 16, wherein said first memory region is divided into first and second sub-regions, and wherein a function call from the second memory region is permitted into only one of said first and second sub-regions.

23. The system of claim 16, wherein at least a portion of device driver code is located in the second memory region.

24. The system of claim 16, wherein at least a portion of trusted application code is located in the first memory region.

25. A computer program product comprising program instructions embodied on a media, wherein said instructions, when loaded into a computer system running one or more application programs and an operating system incorporating a kernel, cause the computing system to perform the method of:

providing a privileged mode for executing routines associated with said kernel, and a non-privileged mode for executing routines associated with an application program;

locating the kernel within a first region of memory, and the application program in a second region of memory; and

setting the system to privileged mode in response to accessing code in said first region of memory and to non-privileged mode in response to accessing code in said second region of memory.

26. The computer program product of claim 25, wherein the application program in the second memory region accesses the kernel in the first memory region by making a system call.

27. The computer program product of claim 26, wherein said system call is implemented as a standard function call.

28. The computer program product of claim 27, wherein said setting comprises:

switching to privileged mode in response to a function call from the second memory region into the first memory region; and switching to non-privileged mode in response to a function call from the first memory region into the second memory region.

29. The computer program product of claim 27, wherein said setting further comprises reverting to a previous mode at the conclusion of a function call.

30. The computer program product of claim 27, wherein said setting further comprises comparing a calling address and a called address for the function call against predetermined address limits to determine in which memory region said calling address and called address are located.

31. The computer program product of claim 25, wherein the computer system memory is divided into pages, and the program instructions further cause the computing system to associate an indicator with each page representative of whether the page is located in said first region of memory or said second region of memory.

32. The computer program product of claim 31, wherein said indicator is stored in a page translation table.

33. The computer program product of claim 31, wherein said indicator has a first value for the second memory region, and a second or third value for said first memory region, and wherein:

code on a page having said first value executes in non-privileged mode and can be accessed from code on a page having said first, second or third value;

code on a page having said second value executes in privileged mode and can be accessed from code on a page having said first, second or third value; and

code on a page having said third value executes in privileged mode and can be accessed from code on a page having said second or third value.

34. The computer program product of claim 33, wherein a page having said second value is used to reference code routines on one or more pages having said third value.

35. The computer program product of claim 25, wherein said first memory region is divided into first and second sub-regions, and wherein a function call from the second memory region is permitted into only one of said first and second sub-regions.

36. The computer program product of claim 25, wherein said first and second memory regions are determined as part of system initialisation.

37. The computer program product of claim 25, wherein at least a portion of device driver code is located in the second memory region.

38. The computer program product of claim 25, wherein at least a portion of trusted application code is located in the first memory region.